Online Spatio-Temporal 3D Convolutional Neural Network for Early Recognition of Handwritten Gestures

William Mocaër, Eric Anquetil, Richard Kulpa

Online Spatio-Temporal 3D Convolutional Neural Network for Early Recognition of Handwritten Gestures*

William Mocaër[1,2], Eric Anquetil[1], and Richard Kulpa[2]

[1] Univ Rennes, CNRS, IRISA
[2] Univ Rennes, Inria, M2S
F-35000 Rennes, France
firstname.lastname@irisa.fr

Abstract. Inspired by recent spatio-temporal Convolutional Neural Networks in computer vision field, we propose OLT-C3D (Online Long-Term Convolutional 3D), a new architecture based on a 3D Convolutional Neural Network (3D CNN) to address the complex task of early recognition of 2D handwritten gestures in real time. The input signal of the gesture is translated into an image sequence along time with the trajectory history. The image sequence is passed into our 3D CNN OLT-C3D which gives a prediction at each new frame. OLT-C3D is coupled with an integrated temporal reject system to postpone the decision in time if more information is needed. Moreover our system is end-to-end trainable, OLT-C3D and the temporal reject system are jointly trained to optimize the earliness of the decision. Our approach achieves superior performances on two complementary and freely available datasets: ILGDB and MTGSetB.

Keywords: Spatio-Temporal Convolutional Neural Network · Early recognition · Handwritten gesture · Online Long-Term C3D · WaveNet 3D.

1 Introduction

To be reactive, some applications need to know as soon as possible the intention of the user. In a tactile environment where you can zoom, scroll with direct manipulation, we should also be able in the same interactive context to do more complex actions associated with real gestures like symbols (abstract command). The coexistence of direct manipulation and abstract command is possible in interactive context only if we are able to predict very early the intention of the user, before the gesture is completed. Very few works addressed this coexistence problem, but we can find the ones of Petit & Maldivi [15] and Kurtenbach & Buxton [10]. Most existing works focus on the recognition of gestures once it is completed, only few covered the early recognition problem which is a complex

* This study is funded by the ANR within the framework of the PIA EUR DIGISPORT project (ANR-18-EURE-0022).

new challenge for the 2D handwritten gesture recognition community, but also for the 3D human gesture recognition community.

In most of the cases, it is possible to discriminate the gesture before it is completed. We define the early recognition problem as the task to predict the class of a gesture as soon as possible. To avoid errors the system should be able to postpone the decision if more information is needed, which is linked to confidence. Early gesture recognition opens a large field of new applications with the coexistence of direct and abstract commands in the same context.

To tackle the aforementioned challenges, we propose the novel OLT-C3D network, coupled with an integrated temporal reject option system. Inspired by recent spatio-temporal CNNs [3,17] in computer vision field, we propose a new architecture based on 3D CNN. The input signal of the gesture is translated into a sequence of images along time with the cumulative trace. Then the images are passed into the 3D CNN which gives a prediction at each new frame. We provide to OLT-C3D a capacity of auto-evaluation of the prediction thanks to the temporal reject system: the prediction can be either accepted to confirm the prediction or rejected if the network needs to wait for more information to decide. The main contributions of this paper are summarized as follows:

- We designed a representation strategy to translate the online input signal in free context into an image sequence.
- We propose the OLT-C3D network, a 3D convolutional neural network built to handle 2D image along time in an online manner.
- We added a temporal reject option system to the OLT-C3D network to postpone the decision in time if more information is needed.
- The network is end-to-end trainable and do not need any post-calibration for the reject system.
- Our method achieves superior performance for the early recognition task regarding accuracy and earliness. Experiments were conducted on two freely available and complementary datasets: ILGDB [16] (mono-stroke gestures) and MTGSetB [5] (multi-touch gestures).

2 Related Works

We believe that the task of 3D gesture recognition is very close to the recognition of 2D gesture, and particularly the recognition from skeleton joints. The 3D gesture recognition community is particularly active these last years. Most work focuses on trimmed gesture recognition, some works addressed the untrimmed gesture recognition and only few tackle the early recognition problem. In 2D, the early recognition of handwritten gesture is also very few addressed. We present in this section the works related to the 3D and 2D early gesture recognition and prediction.

In 3D, some works addressed the early recognition task with template-based methods. For example, Kawashima et al. [8] proposed a method that computes the distance between input gestures and templates. The system prediction is

rejected until the distance with the second most similar class template is over a threshold. Mori et al. [13] used a partial matching method to do early recognition and gesture prediction. Bloom et al. [1] also proposed an early action recognition and prediction method based on template matching using DTW. More recently, some works proposed deep learning-based methods. A system based on a recurrent 3D CNN (R3DCNN) proposed by Molchanov et al. [12] is able to do early recognition, the input video is split in clips and passed into a 3D CNN, then the output is given as input of an RNN. They used a reject system based on the confidence score emitted by the classifier and a fixed threshold. Weber et al. [19] used LSTM network with the 3D joints coordinates in input, the reject strategy consists of waiting that the system repeats the same class prediction a fixed number of consecutive frames. Boulahia et al. [2] proposed a more explicit and transparent method based on a combination of curvilinear models, they indexed the gesture completion using displacement instead of time in order to be speed-independent in the gesture representation. They used a reject system based on the confidence score given by an SVM.

Escalante et al. [6] and Liu et al. [11] addressed the task of action prediction of 3D gestures in an untrimmed stream. Both proposed a method to predict the class at any observation ratio of the gesture without any reject option. Escalante et al. method is based on naive Bayes. Liu et al. developed an architecture named SSNet based on WaveNet [14]. They used a hierarchy of stacked causal and dilated 1D convolutional layers. SSNet is able to handle a stream in real time, giving a new response to each new frame, but no reject system is incorporated to the method. Some of these methods can be adapted to the early recognition of 2D gestures.

Few works addressed the task of early recognition of **2D gestures**. Uchida et al. [18] proposed an early recognition system based on frame-classifier combination. A frame classifier at time t uses weighted combination of the previous $1 \dots t-1$ frame classifiers. Chen et al. [4] addressed the task by using a combination of length-dependent classifiers and a system of reject based on the confidence scores of the classifiers and repetition of prediction. Recently, Yamagata et al. [20] proposed an approach to do handwriting prediction which learns the bifurcations of gestures based on an LSTM network.

The 3D gesture recognition methods particularly inspired us to develop our approach to address the early recognition of 2D gestures task.

3 Method

Firstly, inspired by trajectory-based methods [1,2,18], we propose an original **spatio-temporal representation**, representing the gesture completion in time. The online signal of the gesture is translated into an image sequence containing the trajectory, each new image of the sequence is incremented by a new piece of the trajectory. Then, the images are passed into our **OLT-C3D network**, this original network is mainly inspired by recent spatio-temporal CNNs [3,17] which have proven their abilities to learn spatio-temporal features. OLT-C3D gives a

prediction at each new frame. Finally, the **temporal reject option system** is able to postpone the decision by rejecting the predictions.

3.1 Spatio-Temporal Gesture Representation Strategy

In this work we present an original spatio-temporal representation of the gesture. The input signal is an online trajectory, at each instant we know the position of fingers/pen on the device. We will translate this input signal into an image sequence, each new image containing the new positions of the fingers with its previous trajectories. This representation will be the input of the spatio-temporal CNN. At each instant we will feed the CNN with the new information received in order that the input is always up to date. At the beginning, the network sees only a small piece of the gesture, at the end, it sees the full gesture trace with all the history of the gesture completion. The history is very important to see in which order the gesture is done: two gestures can have the same final shape, but are not done is the same order, this is illustrated in figure 1.

A naive strategy would be to feed the neural network with a new image containing the new information at each time we have new information from the device used, most of the works in 3D gesture recognition [1,11,12,17] use this strategy. Nevertheless between each time there can be only a very small amount of new information if the gesture is done slowly, or even no new information at all if the gesture is paused. By translating the input signal into a new image at each instant, this would lead to a lot of images with some duplication and a very small amount of new effective information between images. Furthermore, if we choose a larger sample rate to reduce the number of images, a gesture made slowly and the same gesture done quickly would not produce the same amount of images along time. For the quick gesture, we would not be able to recognize the gesture as soon (in terms of quantity of information) as we would be able to recognize. To tackle these problems we used, as previous studies [2,4], the quantity of information (displacement) instead of the time to quantify the effective displacement and generate a speed independent image sequence. Each new image will be incremented by the same total displacement θ. A smaller θ will lead to more images than a higher one, because the displacement between each new image will be smaller we need more images to complete the gesture. This strategy leads to three main advantages: fewer images in the sequence, more significant difference between each image and speed invariant representation. For simplicity, we use the term "frame" to designate each image incremented by a certain amount of displacement.

The network has all the history of the trace, but it cannot make the difference between a simple touch then finger up and a constant touch. We need a strategy to make this difference appear on our representation, this is necessary for some gestures in multi-touch (i.e., multiple fingers are used to make a gesture) databases like MTGSetB [5]. To this end, we add a second channel to our images, containing a "1" value if a finger is active in this coordinate at the end of the current displacement, a null value otherwise. This leads to two images: one containing the trajectory, and the other one, very sparse, containing ones

where a finger is active. These two images will be used as channels by the CNN. This second channel has been used only for the multi-touch dataset MTGSetB. For mono-touch datasets such as ILGDB, it has no benefits.

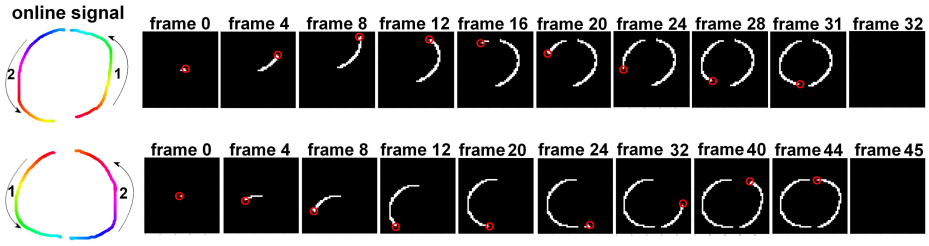


Fig. 1: Considering the final shape, two gesture classes can be involved regarding the order of the stroke. Regarding the second gesture, the order of the strokes is reversed from the first gesture. Our representation takes in consideration the order of the strokes thanks to the history. The last finger position in the segment is represented by a red pixel (surrounded by a red circle in images for visibility) in our representation. If the gesture reaches the border of the image, all the gesture is shifted in new images (this is visible in both gestures between frame 8 and 12). The end of the gesture is symbolized by a black image.

Another difficulty is the *free context*: we don't know in advance what will be the size of the full gesture, so we cannot ensure that the gesture will enter correctly in the image dimension with respect to the resolution we have at the begin of the draw. One solution is to rescale the gesture at each new frame to fit the initial dimension, but we think that would break the spatio-temporality of the information that will be used by the CNN, with difficulties to perceive which new piece is just added between two frames. To keep the same scale from the begin to the end, we choose to "follow" the movement by shifting all the gesture in the opposite direction of the movement when it reaches the edge of the image. We potentially lose a piece of the gesture at each time we shift, but the network does not need it anymore because this part is still in its history.

Lastly, in a database where some gestures are subpart of others, the network needs to know when the gesture is finished. To do that, we add a black image at the end of the gestures. In real application, a strategy must be established to determine when a gesture is finished, it can be pen up from the device, explicit confirmation, time without any action... The final representation is illustrated in figure 1.

3.2 Online Spatio-Temporal 3D CNN with Temporal Reject System

Recently, the CNNs have proved their ability on learning from time series [11,14] and image sequences [3,17]. Inspired by these approaches, we propose OLT-C3D

(Online Long-Term Convolutional 3D), a spatio-temporal 3D CNN able to treat streaming data from devices and to give a response in real time.

Online Spatio-Temporal 3D CNN. We propose a new architecture mainly inspired by two networks: WaveNet [14] and C3D [17]. WaveNet is able to handle 1D temporal series in an online manner using causal and dilated convolutions. C3D is a 3D CNN able to handle 3D input: 2D images along time.

Our objective was to be able to handle 3D input in an online manner. To do that, we propose a spatio-temporal 3D CNN with causal and dilated convolutions on the temporal axis. On the spatial dimensions, the network follows standards using alternatively convolutional layer and pooling layer. An example of a 3D convolution (first layer) of OLT-C3D is provided in figure 2.

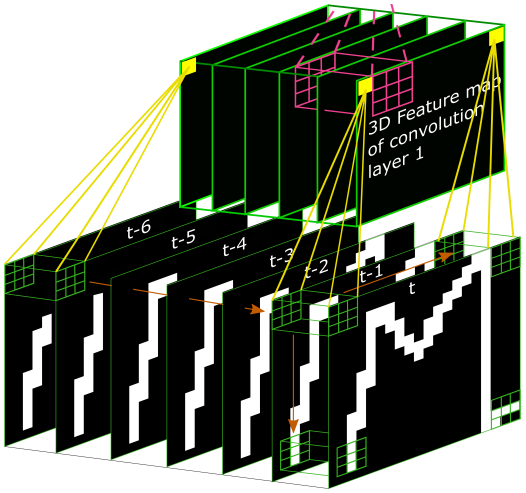


Fig. 2: Example of a spatio-temporal 3D convolution in the first layer. The filter size is 2 (time) $\times$ 3 (x axis) $\times$ 3 (y axis). The filters are applied along all the temporal axis with causal convolutions, and along spatial dimensions with classic convolutions. There is no dilatation along temporal axis for the first layer. The green filter is one of the first layer. The pink filter is one of the second layer. The filters can learn spatio-temporal patterns thanks to the 3D convolution.

Our architecture is composed of 10 stacked convolutional layers. The convolutions are causal on temporal dimension. The causal convolutions let the network compute an output only from previous frames, this ensure not to use future frames for the predictions. The layers also use dilated (or "à trou") convolutions along the time axis. The dilated convolutions allow the network to grow its receptive field quickly with the layers, ensuring that all the history of the frames is used to compute the new output. These layers are divided in two blocks of 5 layers, with a dilatation rate equals to 2^i where $i \in \{0, 1, 2, 3, 4\}$ is the index of

the layer in the block, the dilatation rate is 16 for last layers of blocks. Taking these dilation rates allows to increase the receptive field and to assure that all initial values given in input are taken into account. Each convolution layer is followed by a max-pooling layer applied to the two spatial dimensions, there is no pooling along the temporal dimension.

As shown in figure 3, the lower convolutional layer has a very small receptive field since only two frames are used to compute the output, it focuses on the last two frames. The second layer uses the result of the previous one, using indirectly four frames. The number of frames used increases with the number of layers. The top convolutional layer of the network is able to see up to 63 input frames. The receptive field must be accorded to the parameter θ defined in section 3.1 and the length of gestures. With the θ we used in our experiment, 63 frames in enough to see the whole gesture completion history in most cases. These dilated convolutions allow our system to avoid the use of memory cell mechanisms like RNN used in some works [12,19] to have a long-term memory.

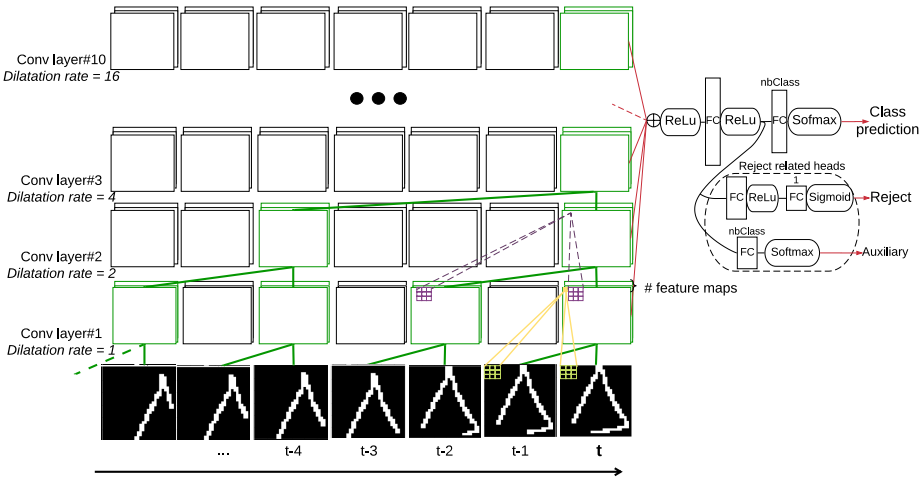


Fig. 3: The complete architecture of OLT-C3D. First, the signal is transformed into images, and fed into the network. The network makes a prediction at each instant. OLT-C3D is coupled to a temporal reject system.

The network emits a classification prediction for each new frame. To this end, the network is completed by an aggregation (average) of the feature maps from all layers corresponding (thanks to causal convolutions) to **the last frame (instant t)**. Because the network only uses the feature maps corresponding to the last frame, our network can handle any length of sequence and is able to give a prediction at each new frame. The aggregation is followed by a fully connected (FC) layer and then three main heads. The first head f is dedicated to the class prediction, composed of a new FC layer with the number of class

neurons followed by a softmax activation function, the second head (Reject, g) and third head (Auxiliary, h) are dedicated to the reject option system. The complete architecture is provided in figure 3.

Temporal Rejection Option System. One main problem in early recognition is to take a decision to recognize a gesture only when the answer is sure, we want to let the classifier the possibility to postpone a decision when it estimates that it does not have enough information. For example, if we have gestures with common parts at the beginning, we want the classifier to give no response until the common part is passed. We need a mechanism in order to have a kind of confidence score, to reject or accept the current prediction.

To tackle this problem, we used SelectiveNet [7] and adapted it to a per-frame fashion. This leads us to add two new outputs: selection/reject head and auxiliary head, these two heads are included in the "Reject related heads" block shown in the figure 3. The reject head is composed of a FC layer with ReLu activation. Then it lasts by a FC layer with only one neuron. Sigmoid is the final activation function used in this head, we define this output as g. The goal of this output is to accept or reject the prediction. We will consider the prediction rejected if the reject head output is less than a parameter τ, and accepted if it is over. The final prediction output with respect to g is defined as:

$$(f, g_\tau)(x) = \begin{cases} f(x), & \text{if } g(x) \geq \tau \\ \text{don't know}, & \text{otherwise.} \end{cases} \tag{1}$$

We used $\tau = 0.5$ in this paper, as in SelectiveNet. Finally, the loss of the prediction head, using g, is defined as follows:

$$\mathcal{L}_{(f,g)} = \frac{1}{m} \sum_{i=1}^{m} \ell\left(f\left(x_i\right), y_i\right) g\left(x_i\right) + \lambda \Psi\left(c - \hat{\phi}\left(g\right)\right) \tag{2}$$

where c is the target coverage, λ is a hyperparameter relative to the importance of the coverage constraint and $\Psi(a) = \max(0, a)^2$. As SelectiveNet we used $\lambda = 32$. $\hat{\phi}(g)$ is the empirical coverage, i.e. the average value of $g(x)$, and ℓ is the cross entropy loss.

The auxiliary head is the same as the prediction head, but is optimized with a standard loss function $\mathcal{L}_h$, we used cross-entropy. It is used to optimize the CNN representation without focusing too much on the loss of the prediction head $\mathcal{L}_{(f,g)}$. More details about SelectiveNet can be found in the original paper [7].

Unlike in SelectiveNet where the rejection head is responsible for the rejection of a full-gesture sample, here the rejection head has to decide for each frame if the prediction is accepted. The loss has to be adapted from a per-sample fashion to a per-frame fashion. In order to do this, the coverage average $\hat{\phi}(g)$ and the loss ℓ are computed for all predictions along time. Consequently, the coverage c is more related to earliness in our case.

The final optimized loss is:

$$\mathcal{L} = \alpha \mathcal{L}_{(f,g)} + (1 - \alpha) \mathcal{L}_h \tag{3}$$

in our experiments we fixed α to 0.5 as SelectiveNet.

At each instant OLT-C3D will output one class prediction and the temporal reject system accepts or rejects the prediction if it needs more information. The network can also totally reject the gesture, even at the end, if the gesture is too close to two classes, or if it does not correspond to a known class.

4 Experimental Evaluation

We evaluate the OLT-C3D approach on two freely available datasets: ILGDB [16] which contains only **mono-stroke** gestures and MTGSetB [5] which contains **multi-touch** gestures. These two datasets are complementary in terms of gesture natures (mono/multi-stroke, mono/multi-touch), they are very interesting for early recognition experiments. We compare our scores to the state-of-the-art method on the task of early recognition [4] on these two datasets.

4.1 Network Hyperparameters and Details

The pooling size and stride of the maxpooling layers is 3 for spatial dimensions, 1 for temporal dimension (no pooling). We used a small dropout for convolutional layers of 0.1, and 0.3 for the first FC layer. ReLu is the activation function used after each convolutional layer. We optimized the network with Adam [9] with the learning rate fixed to 0.003. 85 % of the training data is used for training, 15 % is used as validation set. The dataset specific hyperparameters are presented in table 1. For ILGDB, we fixed the image dimension to 30 by 30. The coordinates of the gestures are scaled by 0.2, and we augmented the training data by scaling the train gestures coordinates also by 0.3, 0.4, 0.5 and 0.6. The displacement quantity θ is fixed to 4.5 pixels (once scaled). Regarding the CNN, we found that 10 filters by layers is enough for this dataset, with 300 neurons in the FC layers. We used a batch size of 85 sequences padded with black images at the end. The target coverage c is fixed to 0.6. The hyperparameters for MTGSetB leads to a higher network (770k parameters for MTGSetB and 420k for ILGDB), this is because this dataset is more complex than ILGDB due to a higher number of gesture classes and shape varieties. These hyperparameters have been softly fine-tuned with a validation set.

Table 1: Dataset specific hyperparameters.

	Image dim.	Scale	Data aug. scale	filters	θ	FC neurons	Batch size	c
ILGDB	30x30	0.2	0.3, 0.4, 0.5, 0.6	10	4.5	300	85	0.6
MTGSetB	40x40	0.03	0.04	25	2	150	40	0.75

4.2 Measures

To evaluate the early recognition task, we use the True Acceptance Rate (TAR) which measure the accuracy of the classifier when the prediction is accepted. We

also use the False Acceptance Rate (FAR) which measure the error rate when the prediction is accepted. To measure the final reject, we use the Reject Rate (RR) which is the number of samples which are totally rejected, even at the end of the sequence. Referring to the notation of the Table 2 the TAR, the FAR and RR are defined as:

$$TAR = \frac{N_A^T}{N} \ ; \ FAR = \frac{N_A^F}{N} \ ; \ RR = \frac{N_R}{N} \tag{4}$$

Regarding our network, only the classification of the first acceptation is used to compute the TAR and the FAR. If the gesture is never accepted, it is taken into account in the RR. Note that $TAR + FAR + RR = 1$. To evaluate the earliness, we used the Normalized Distance To Detection (NDtoD) which is computed as the length of the gesture made at the moment of the first acceptation over the total length of the gesture, only the accepted gestures are taken into account.

Table 2: Notations used for the measurement of reject-based systems, used in [4]. N_A^F are the training samples which are wrongly classified but accepted by the reject system while the N_A^T samples are correctly classified and accepted.

Sample set (N)	Reject option	
	Accept (N_A)	Reject (N_R)
Correctly classified (N_{cor})	True Accept (N_A^T)	False Reject (N_R^F)
Mis-classified (N_{err})	False Accept (N_A^F)	True Reject (N_R^T)

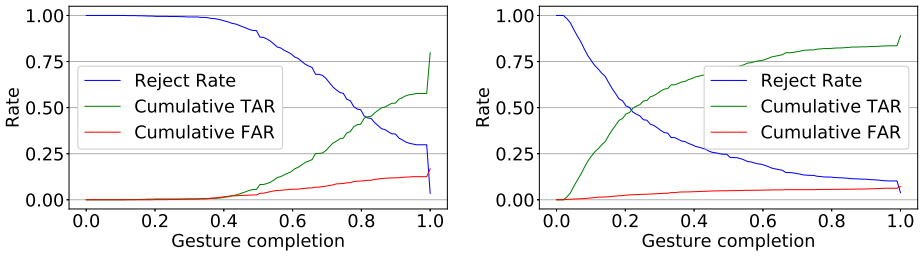
4.3 Early recognition results

ILGDB. The ILG database is a **mono-stroke pen-based gestures dataset** performed by 38 users. It contains 21 different gesture classes with a total of 1923 samples, 693 are used for training and 1230 for testing. The specificity of this dataset is that there are a lot of gestures which have common begins, or are subparts of other gestures. With this, the network needs to reject until the trajectory become discriminative, and it can be very late. We compared our score to the approach of Chen et al. [4] using the predefined Train/Test split furnished by the dataset. To compare fairly with them, we used the parameter t which is the number of time the same prediction class must be accepted consecutively by our reject system to be finally accepted. This is a way to reinforce the confidence, but it leads to delayed decisions and a higher reject rate, which can be less optimal than tuning the reject system. $t = 1$ is the default value for other experiments, unless explicit mention. The scores are in table 3. We can see that for an equivalent earliness ($t = 2$ for Chen et al. and $t = 1$ for us), our network is much more accurate with 15 % of better classification when the gesture is accepted, 10 % less misclassification, and fewer gestures rejected. The decision is made on average at 76 %, which is late but consistent with the gestures.

Some gestures have common begins, for these the network needs to wait that the common part is passed. In figure 4a, we see that the network reject most of the prediction in early states, waiting that the gesture completion is at least 40 % to begin to accept the predictions. The FAR stay low until the end. We can see a peak at 100 % of the gesture completion, that is because some gestures are subpart of others, so the only way to recognize the gesture is to wait that the gesture is completely finished.

Table 3: Comparison of the OLT-C3D approach to the previous method on the datasets. TAR: True Acceptance Rate, FAR: False Acceptance Rate, RR: Reject Rate, NDtoD: Normalized Distance to Detection (earliness). t is the number of consecutive acceptation of the same class required to be finally accepted.

Dataset	t	Chen et al. [4]				**OLT-C3D (ours)**			
		TAR	FAR	RR	NDtoD	TAR	FAR	RR	NDtoD
ILGDB	1	30.65 %	67.15 %	2.20 %	34.81 %	**79.75** %	**16.74** %	**3.50** %	**76.81** %
	2	**64.15** %	**26.42** %	**9.43** %	**75.53** %	81.79 %	14.15 %	4.07 %	82.56 %
	3	73.98 %	11.22 %	14.80 %	92.24 %	83.25 %	12.03 %	4.72 %	87.44 %
	4	77.72 %	6.26 %	16.02 %	97.62 %	84.31 %	9.51 %	6.18 %	91.01 %
	5	77.80 %	4.88 %	17.32 %	99.19 %	85.20 %	7.40 %	7.40 %	93.71 %
	6	77.72 %	4.55 %	17.72 %	99.68 %	84.39 %	6.34 %	9.26 %	95.54 %
MTGSetB	1	**81.89** %	**14.56** %	**3.54** %	**37.04** %	89.25 %	7.24 %	3.51 %	30.77 %
	2	83.44 %	10.85 %	5.71 %	46.82 %	**90.52** %	**5.82** %	**3.66** %	**36.89** %
	3	82.38 %	8.85 %	8.77 %	55.89 %	90.94 %	5.08 %	3.98 %	42.78 %
	4	82.20 %	6.06 %	11.73 %	66.16 %	91.22 %	4.25 %	4.53 %	48.48 %
	5	80.35 %	4.60 %	15.05 %	71.03 %	91.49 %	3.53 %	4.98 %	53.92 %
	6	77.42 %	3.41 %	19.17 %	77.54 %	91.36 %	2.86 %	5.77 %	58.92 %

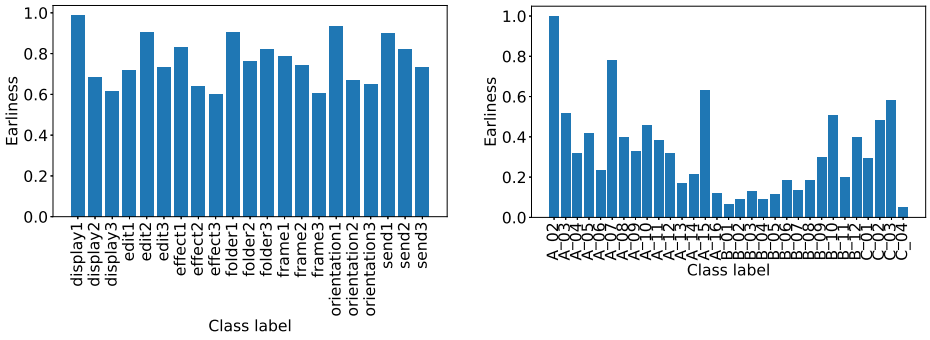


(a) ILGDB (b) MTGSetB

Fig. 4: Behaviour of each rate (Cumulative TAR, cumulative FAR, RR) on the two datasets. The cumulative TAR at x% of completion is the number of samples accepted before x% and correctly classified over the total number of samples.

MTGSetB. The MTGSetB dataset is composed of 45 different **multi-touch gestures** regrouped into 31 rotation invariant gesture classes made by 33 users. Like ILGDB, we compared our score to Chen et al. [4]. 50% of the data are used for training. The comparison is shown in the table 3. We can see that, for the same earliness and reject rate ($t = 1$ for Chen et al. and $t = 2$ for us), OLT-C3D has a much higher TAR (+9%), a much lower FAR (−9%). The MTGSetB dataset contains multi-touch gestures, and only few of them are subparts of others. By using the number of fingers and the initial directions, our network is able to predict the gestures very early, with few errors. In figure 4b, we see that the network starts to accept very early, keeping a low FAR.

Earliness Evaluation per Class. The earliness per-class is presented in figure 5. For ILGDB, we observe differences between classes, with earliness from an average of 60 % gesture completion for the class "effect3" to 100 % for the class "display1". Note that the gesture corresponding to "display1" is a subpart of two other gesture classes, so the only way to recognize the gesture correctly is to wait that the gesture is completely finished. In MTGSetB we observe a similar case with the class "A_02" which is a subpart of multiple other gesture classes. Otherwise, the gestures are accepted very early for most of the classes. The earliest predicted class is the last one, "C_04", which is accepted on average at 5 % of gesture completion. The network is able to predict as early for this class because it is the only one with only one touch which begins by moving to the right, so there is enough information in the first frame in most cases.



(a) ILGDB (b) MTGSetB

Fig. 5: Earliness per class (NDtoD)

4.4 Spatio-Temporal Representation Evaluation

We compare our representation discussed in section 3.1 with two variants: one with only the trajectory, the other with only the finger position. The results

are provided in table 4. We see that both trajectory and finger position bring significant information to the representation. In particular, the finger position channel brings the difference between a constant touch and a touch then release.

Table 4: Comparison of the different variants of our representation on MTGSetB.

Variant	TAR	FAR	RR	NDtoD
Only Trajectory (first channel)	84.1 %	9.73 %	6.17 %	**30.71** %
Only Finger Position (second channel)	86.54 %	10.87 %	2.59 %	33.28 %
Both: Trace and Finger Position	**89.25** %	**7.24** %	3.51 %	**30.77** %

4.5 Qualitative Results

In the datasets, there are some gestures with common parts. The expected behavior from our network is to reject the prediction until the common part is passed. In this section, we analyze the output of our system. For example, ILGDB contains three gestures which starts like an "M" letter, the direction given after the "M" stroke is decisive. The figure 6 shows an example of the behavior of our network on these three labels. We see that the temporal reject system waits the decisive instant to accept the prediction. Note that this example is very representative of the behavior of the network on the "M" classes. This shows the ability of our approach to well reject in time the prediction until the common part is passed. The case of the "display1" class is also interesting, this gesture is a simple down stroke, as illustrated in figure 6. This gesture is a subpart of two other gestures, where one goes left and the other go right after the down stroke. The only way the network can know that it is the class "display1" is to wait until the end of the gesture (e.g., pen up for single stroke gestures). As explained in section 3.1 we modeled the end of the gesture using a black image. For this gesture class the network rejects the prediction until the black image.

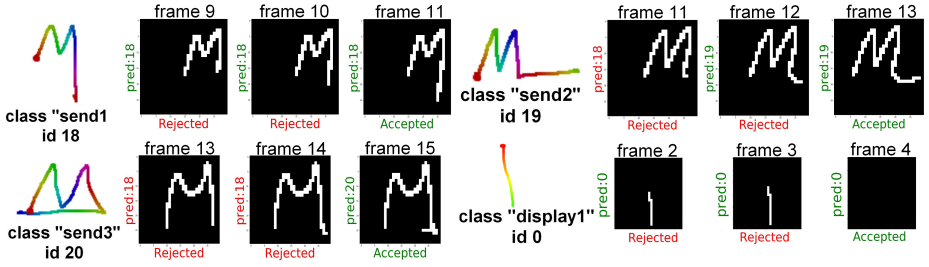


Fig. 6: Behavior on the "M" classes until the first acceptation. The system rejects predictions until the decisive instant. The class "display1" is a subpart of two other classes, the system rejects the predictions until the black frame (end of the gesture). A green label prediction on the left means a good classification.

On MTGSetB, the first acceptation is made very early on average. By analyzing the number of touch and the beginning of the trajectory, the network is able to make accurate predictions and very early. An example is shown in figure 7. In this example, we see that the only element which makes the difference between the two gestures is the finger position channel. Without this channel, the network would not be able to discriminate these two gestures.

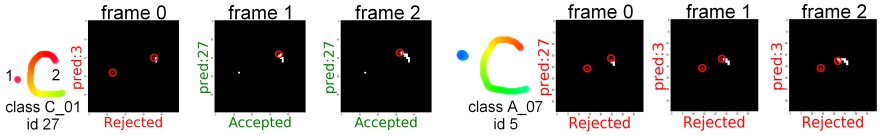


Fig. 7: Behavior on the classes C_01 and A_07 from MTGSetB. For the gesture on the left, the network is able to accept the prediction in the frame 1 because it can see that the finger of the left stroke has been released, and it is the only gesture which starts like that. For the gesture on the right, the finger of the left stroke is continuously pressed, that can be multiple gestures at this instant, so it rejects the predictions for these frames.

4.6 Speed Execution

Our network is able to treat streaming data at **46** frames per seconds, considering that the frame representation extraction time execution is negligible. Note that our system waits to acquire enough displacement to submit the new image to the network, in this case it should be able to respond in ≈**22ms**. This is enough to be used in real-time application. Experiments were conducted on a Quadro RTX 3000. The response time can be improved using the activation sharing scheme used in SSNet [11].

5 Conclusion

In this paper, we proposed a framework composed of three main parts. First, an original **spatio-temporal representation** of the gesture, well suited to represent online gesture, regardless of its nature (mono/multi-stroke, mono/multi-touch). Then, **OLT-C3D**, an original 3D CNN able to extract spatio-temporal features in an online manner. Lastly, a **temporal reject system** to postpone the decision if necessary. Our network coupled to the temporal reject system is end-to-end trainable, and runs in real time. We showed that our method is able to make predictions very early, with very interesting performance. It opens a large field of innovative applications. Our future works will focus on an extension of this approach for early recognition of 3D gestures.

References

1. Bloom, V., Argyriou, V., Makris, D.: Linear latent low dimensional space for online early action recognition and prediction. Pattern Recognition **72**, 532–547 (2017). https://doi.org/10.1016/j.patcog.2017.07.003
2. Boulahia, S.Y., Anquetil, E., Multon, F., Kulpa, R.: Détection précoce d'actions squelettiques 3D dans un flot non segmenté à base de modèles curvilignes. In: RFIAP 2018 Reconnaissance des Formes, Image, Apprentissage et Perception. pp. 1–8. Paris, France (Jun 2018)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017). https://doi.org/10.1109/CVPR.2017.502
4. Chen, Z., Anquetil, E., Viard-Gaudin, C., Mouchère, H.: Early recognition of handwritten gestures based on multi-classifier reject option. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 01, pp. 212–217 (2017). https://doi.org/10.1109/ICDAR.2017.43
5. Chen, Z., Anquetil, E., Mouchère, H., Viard-Gaudin, C.: Recognize multi-touch gestures by graph modeling and matching. In: 17th Biennial Conference of the International Graphonomics Society. Drawing, Handwriting Processing Analysis: New Advances and Challenges, International Graphonomics Society (IGS) and Université des Antilles (UA), Pointe-a-Pitre, Guadeloupe (Jun 2015)
6. Escalante, H.J., Morales, E.F., Sucar, L.E.: A naïve bayes baseline for early gesture recognition. Pattern Recognition Letters **73**, 91 – 99 (2016). https://doi.org/10.1016/j.patrec.2016.01.013
7. Geifman, Y., El-Yaniv, R.: Selectivenet: A deep neural network with an integrated reject option. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2151–2159. PMLR (09–15 Jun 2019)
8. Kawashima, M., Shimada, A., Nagahara, H., Taniguchi, R.: Adaptive template method for early recognition of gestures. In: 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV). pp. 1–6 (2011). https://doi.org/10.1109/FCV.2011.5739719
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
10. Kurtenbach, G., Buxton, W.: Issues in combining marking and direct manipulation techniques. In: Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology. p. 137–144. UIST '91, Association for Computing Machinery, New York, NY, USA (1991). https://doi.org/10.1145/120782.120797
11. Liu, J., Shahroudy, A., Wang, G., Duan, L., Kot, A.C.: Skeleton-based online action prediction using scale selection network. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(6), 1453–1467 (2020). https://doi.org/10.1109/TPAMI.2019.2898954
12. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J.: Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016). https://doi.org/10.1109/CVPR.2016.456
13. Mori, A., Uchida, S., Kurazume, R., Taniguchi, R., Hasegawa, T., Sakoe, H.: Early recognition and prediction of gestures. In: 18th International Conference on Pattern Recognition (ICPR'06). vol. 3, pp. 560–563 (2006). https://doi.org/10.1109/ICPR.2006.467

14. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. CoRR (2016)
15. Petit, E., Maldivi, C.: Unifying gestures and direct manipulation in touchscreen interfaces (12 2013)
16. Renau-Ferrer, N., Li, P., Delaye, A., Anquetil, E.: The ilgdb database of realistic pen-based gestural commands. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012). pp. 3741–3744 (2012)
17. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497 (2015). https://doi.org/10.1109/ICCV.2015.510
18. Uchida, S., Amamoto, K.: Early recognition of sequential patterns by classifier combination. In: 19th International Conference on Pattern Recognition. pp. 1–4 (2008). https://doi.org/10.1109/ICPR.2008.4761137
19. Weber, M., Liwicki, M., Stricker, D., Scholzel, C., Uchida, S.: Lstm-based early recognition of motion patterns. In: 2014 22nd International Conference on Pattern Recognition. pp. 3552–3557 (2014). https://doi.org/10.1109/ICPR.2014.611
20. Yamagata, M., Hayashi, H., Uchida, S.: Handwriting prediction considering inter-class bifurcation structures. In: 17th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 103–108 (2020). https://doi.org/10.1109/ICFHR2020.2020.00029